

# Fitting Random Effects in Semi-parametric Regression Model with Application to Horse Racing

**CHEUNG Man-Yuen**

A Thesis Submitted in Partial Fulfillment  
of the Requirements for the Degree of  
Master of Philosophy  
in  
Statistics

©The Chinese University of Hong Kong  
July 2003

The Chinese University of Hong Kong holds the copyright of this thesis. Any person(s) intending to use a part or whole of the materials in the thesis in a proposed publication must seek copyright release from the Dean of the Graduate School.





Abstract of thesis entitled:

Fitting Random Effects in Semi-parametric Regression Model with Application to Horse Racing

Submitted by Cheung Man Yuen

for the degree of Master of Philosophy in Statistics

at The Chinese University of Hong Kong in July 2003.

## **Abstract**

In this thesis, semi-parametric regression model with random effects is used to analyze the clustered ranking data. The semi-parametric regression model deals with data in the form of ranks. This thesis aims at modeling this form of data with clustering, and develops a computational algorithm to fit the resulting model.

Random effect is a general way for handling clustered data. We fit the random effects with a subject-specific covariance matrix generated by assuming that each individual has a unique set of regression coefficients, which are distributed around the mean of the population.

The horse racing data of the Hong Kong Jockey Club is used throughout the analysis as an application to the ranked data with clustering. It is possible to develop a profitable horse betting strategy, because the payoffs for the pari-mutuel system are determined totally by the public betting.

In our model, the performances of a specific horse are considered as a cluster and the random effects are specific to each particular horse, which has a physical interpretation of racing horse's ability. We show that, with random effects modeling, the overall fit of the data is improved, compared to no cluster modeling.

## 摘 要

在本篇論文中，含隨機作用的半參數迴歸模型會用作分析群聚（rank排序）數據。半參數迴歸模型普遍應用於排序數據上。本論文目的在於定立含群聚等點的排序數據的模型，以及建立該模型的計算法。隨機作用模型是一個通用的群聚數據處理法。我們把隨機作用放入一個假定每個獨立個體擁有一組特定分佈在總體平均值的迴歸係數的共變異數矩陣中。

模型分析中引用了香港賽馬會的賽馬數據作為含群聚的排序數據的應用。由於賽馬賠率純由公眾投注釐訂，所以一個有利潤的賭博策略可被建立。在我們的模型中，每一匹特定馬匹的表現可視為一個群組，而隨機作用則假定為對應於每匹馬，亦可詮釋為該匹馬的能力。我們証實與無考慮群聚的模型比較，加入隨機作用的模型可達到結果的改進。

# Acknowledgement

I would like to express my sincere gratitude to my supervisor, Prof. Gu Ming-Gao, for his patience, tolerance and guidance during the course of this research program. Also, many thanks to Prof. Li Kim-Hung, Prof. Lau Tai-Shing and Prof. Jing Binyi in my thesis committee, for their insightful opinions for improvements of this thesis. Further, I would like to take this opportunity to express my thanks to my family, my fellow classmates and all the staffs of the Department of Statistics.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Rank Regression . . . . .	1
1.2	Clustering . . . . .	2
1.3	Modeling of the ranked data . . . . .	3
1.4	Application in Horse Racing data . . . . .	4
<b>2</b>	<b>Semi-Parametric Regression Model</b>	<b>7</b>
2.1	Review . . . . .	7
2.2	Parameter Estimation . . . . .	9
<b>3</b>	<b>Random Effects</b>	<b>11</b>
3.1	Definition . . . . .	11
3.1.1	A Simple Estimation Algorithm . . . . .	13
3.2	Metropolis-Hastings Algorithm for Simulating Random Effects . .	14
3.3	EM Algorithms for Maximizing the Likelihood . . . . .	16
3.3.1	Stochastic EM Algorithm . . . . .	17
3.3.2	MCEM Algorithm . . . . .	18
<b>4</b>	<b>Application</b>	<b>20</b>
4.1	Fundamental Variables and Variable Selection . . . . .	21
4.2	Simulation Results . . . . .	23
4.3	Betting Strategies and Comparisons . . . . .	25
<b>5</b>	<b>Conclusions and Further Studies</b>	<b>29</b>

Appendix	31
Bibliography	35

# List of Figures

4.1	Plot of the random effects by simple estimation algorithm	31
4.2	Plot of the random effects by REML algorithm	31
4.3	Plot of the random effects by MCMC algorithm	34

# List of Figures

3.1	Plot of the random effects by simple estimation algorithm . . . . .	14
4.1	Plot of the random effects by SEM algorithm . . . . .	24
4.2	Plot of the random effects by MCEM algorithm . . . . .	24



# List of Tables

4.1	The MLE and SE for selected variables under the multinomial logit model	22
4.2	The MLE and SE for selected variables by SEM and MCEM algorithms	23
4.3	Results for simple betting strategy in 221 races . . . . .	26
4.4	Results for Kelly strategy in 221 races . . . . .	28

# Chapter 1

## Introduction

In our daily life, it is unavoidable for us to come across countless occasions of rankings: Rank of Students in Class, Most Watched TV Shows, Movies, Most Valuable Players, and so on. Moreover, ranking also plays an important role in statistics. Many raw data are collected in the form of ranks, questions involving ranks always appear in different questionnaires. In this thesis, we aimed at modeling multistage ranked data with clusters of the features as in the horse racing data.

### 1.1 Rank Regression

In order to model ranked data, numerous data-analytic techniques and probability models have been developed over the years, especially in non-parametric analysis and in the analysis of judging the ranks of objects. In rank prediction, we may think of using the technique of regression, but it is impossible to apply ordinary regression methods even if a rich family of relevant predictor variables are available. However, if suitable specifications and proper estimation methods of corresponding parameters are introduced, rank regression models can be devel-



oped. This kind of model has been used extensively in medical applications and survival analysis, including the proportional hazards model (Cox, 1972), and the proportional odds model (Bennett, 1983). Applications also exists in many other fields such as employment (Lancaster and Nickell, 1980), and the econometric model (Heckman and Singer, 1984).

## 1.2 Clustering

Besides rank regression model, we consider in this thesis individual observations that are correlated or clustered. To be more specific, we consider a pool, which consists of a large number of objects. During each comparison, a certain number of objects are picked from the pool and ranked. After ranking, they are put back into the pool. That is, the incidence that an object is picked several times may occur, and repeated observations for that object can be recorded. These repeated observations from one particular individual forms a cluster.

Clustering is a commonly used technique in Psychiatry (Pilowsky et al., 1969, depressed patients; Paykel and Rassaby, 1978, suicide attempters), Medicine (Wastell and Gray, 1987, facial pain), Social services (Jolliffe et al., 1982, elderly social services), Market research (Green et al., 1967, clustering the cities), Education (Aitkin, Anderson and Hinde, 1981, cluster teachers by teaching behaviour), and Archaeology (Hodson, 1971, hand axes).

Within a cluster, observations usually share certain unobserved characteristics, in which similar behaviour could be observed. As a result, these observations tend to be correlated. Gu, Sun and Huang (2002) developed a frailty model for cluster analysis similar to the conditions here. However, this frailty model cannot be directly used in the situation we consider. In order to capture the cluster

correlations, we introduce random effects (Laird and Ware, 1982) into the rank regression model. The random effects possess a normal distribution and are fitted with a subject-specific covariance matrix generated by assuming that each individual has a unique set of regression coefficients, which are distributed around the mean of the population.

### 1.3 Modeling of the ranked data

Furthermore, we developed a model to cope with the multistage rank data, as suggested by Plackett (1975) based on the ideas from Luce (1959). In addition to the non-parametric model suggested, predictor variables are also used in modeling the probabilities of ranks of individuals. The probability estimation is parameterized in the form of multinomial logit model, which is a popular model applied to discrete choice problems in marketing and economics. Some illustrative applications of this choice behaviour modeling methodology include the selection of a college (Chapman, 1979; Kohn, Manski, and Mundel, 1976; Punj and Staelin, 1978), a mode of transportation (cf. Domencich and McFadden, 1975), a grocery store (Gensch and Recker, 1979), a shopping center (Chapman, 1980), a home (Li, 1977), an occupation (Boskin, 1974), and an electric utility fuel (Joskow and Mishkin, 1977). This model is chosen as it does not involve computational complexity and can be regarded as an elementary model for further consideration of the random effects fitting. This together forms the basis of the semi-parametric regression model used throughout this thesis.

After fitting random effects into the semi-parametric regression model, the rank of the data can be explained not only by observable factors, but can also be explained by a cluster-oriented unobservable factor (latent variable). This will



lead to a more complete and informative model. With random effects, the model will be more reliable in reflecting the reality.

## 1.4 Application in Horse Racing data

As a heuristic application to the random effects fitted in semi-parametric regression model, the horse racing data of the Hong Kong Jockey Club is used. Betting in horse races is a popular activity in Hong Kong, and the tax obtained from this business plays an important role in government income. This can be shown by the total annual turnover from horse racing for the Hong Kong Jockey Club is more than HK\$70 billion, in which the amount of tax in betting duty is around HK\$10 billion. This amount of tax contributes approximately 7% of the government's annual income. With proper data analysis, we show that a respectable profit can be obtained by making intelligent bets aided by the method discussed in this thesis.

If one bets on horses by randomly picking any horse in a race, the expected return must be negative in the long run. This negative expectation is caused by the pari-mutuel system:

$$Odds_i = \frac{(1 - \rho) \sum_{j=1}^I B_j}{B_i} \quad (1.1)$$

where  $B_i$  is the total amount bet on horse  $i$  in a race of  $I$  horses;  $\rho$  is the track take including tax, which accounts for the negative value expected.

On the other hand, it is also possible to develop a profitable horse betting strategy. This is because the odds given by this system (1.1) depend totally on the public betting, not by the real probabilities of the race outcome. If we can estimate the win probabilities more accurately than the public, a profitable

strategy can be obtained.

Bettors have been searching for profitable wagering systems over the years. It has been proved by academic researchers that with suitable choice of model, horse racing can also be exploited as an investment tool (cf. Vergin, 1977; Ziemba and Hausch, 1984; Ali, 1998; and extensive references). This was motivated by the similarities between the horse racing market and the stock market.

It is interesting to see that horses in a race are ranked according to the place they finished in that race. The absolute running time and how much the first horse is better than the second or the others are not reliable due to the fact that racing surface is changing day to day and the award money is distributed according to ranks. And this can be plugged into the semi-parametric regression model. On the other hand, horses may appear in different races. Each individual horse can then be considered as a cluster, as the quality of a horse is believed to have little changes in different races, especially in a short time span. The random effects are specific to each particular horse. The model can capture the external and internal variations for a horse race, and predict the probability for a particular horse to lead the other horses in a race.

After finding the winning probabilities, we attempt to further progress our studies on obtaining a profitable betting strategy in the horse racing example. This is important for the checking of whether it is valuable to construct this semi-parametric model with random effects and can also justify the accuracy of this rank regression model found. In addition, we can also compare the results obtained by the proposed model with the results of a model without the inclusion of the random effects. It is used to check whether the random effects can improve the model of our interest or not.



This thesis is organized as follows. In Chapter 2 and 3, we provide a more detailed description and the methodology employed to calculate the corresponding parameters for the semi-parametric regression model and the random effects fitting respectively. Further, we also describe the techniques involved in parameters estimation for the model when the semi-parametric regression model and the random effects fitting are combined in the last part of Chapter 3. In Chapter 4, the above model will be illustrated with the real data set of horse racing from the Hong Kong Jockey Club. Finally, conclusions and suggestions for further studies will be given in Chapter 5.

## 2.1 Review

# Chapter 2

## Semi-Parametric Regression Model

In this chapter, a regression model is developed to provide prediction on the probabilities of objects having the observed ranks in one comparison among  $T$  comparisons. Based on a family of relevant predictor variables, we form a rank regression model, which can be used for prediction.

### 2.1 Review

To be an elementary model, a semi-parametric regression model - the Multinomial Logit model is used in this chapter by its computational simplicity for further development in later chapters. This model is established according to the Plackett-Luce model (Plackett (1975), and Luce (1959)), and the model developed from the paper of Gu, Huang and Benter (2002), adopted from the previous model of Chapman and Staelin (1982) and Bolton and Chapman (1986), based on the work of Luce and Suppes (1965) and McFadden (1974).

We consider a random utility function associated with object  $i$ , which is mod-

eled by

$$U_i = \beta' z_i + \epsilon_i, \quad (2.1)$$

in which  $\beta$  is a vector of parameters,  $z_i$  is a vector of predictor variables, or covariates associated with object  $i$ . The residuals term  $\epsilon_i$ , represents the unaccounted effects and random fluctuation of object  $i$  in a comparison.

This utility can be understood as a rough substitute of the magnitude of the response variable. Instead of finding the exact magnitude concerned, the probability of how a particular object is superior to the other objects in a comparison is more relevant to this subject.

The probabilities we are interested in is actually of the form:

$$P_i = \Pr\{U_i > U_j, j = 1, \dots, I; j \neq i\}. \quad (2.2)$$

By assuming the term  $\epsilon_i$  in (2.1) to be independent, with a negative double exponential distribution:

$$\Pr\{\epsilon_i \leq x\} = \exp[-\exp(-x)].$$

The probabilities of interest (2.2) can then be found to have an explicit expression as:

$$P_i = \frac{\exp(\beta' z_i)}{\sum_{j=1}^I \exp(\beta' z_j)}. \quad (2.3)$$

Combining with the Plackett-Luce model, the likelihood function is found to be

$$\begin{aligned} L(\beta) &= \prod_{t=1}^T \Pr\{U_{t(1)} > U_{t(2)} > \dots > U_{t(n)}\} \\ &= \prod_{t=1}^T \prod_{i=1}^n \left\{ \frac{\exp(\beta' z_{t(i)})}{\sum_{j=i}^{I_t} \exp(\beta' z_{t(j)})} \right\} \end{aligned} \quad (2.4)$$

where  $T$  is the total number of comparisons,  $t(i)$  represents the object that has a rank of  $i$  in the  $t^{th}$  comparison, and  $I_t$  represent the number of objects to be



compared in the  $t^{th}$  comparison. Then  $n$  is usually taken as a number smaller than  $I_t$ .

## 2.2 Parameter Estimation

To fit the model to the data in order to solve for the parameter vector  $\beta$ , the Newton-Raphson method (Ralston and Wilf, 1966; Carnahan et al., 1969) is used. By this method, we find  $\beta$  through iterations, in which the results will converge within a few iterations.

We first consider  $I(\beta)$  and  $S(\beta)$ , which are given by

$$I(\beta) = \frac{\partial^2}{\partial \beta^2} \log L(\beta), \quad S(\beta) = \frac{\partial}{\partial \beta} \log L(\beta).$$

From (2.4), we have

$$\log L(\beta) = \sum_{t=1}^T \sum_{i=1}^n \left[ \beta' z_{t(i)} - \log \left\{ \sum_{j=i}^{I_t} \exp(\beta' z_{t(j)}) \right\} \right].$$

Then we can construct

$$\begin{aligned} S(\beta) &= \frac{\partial}{\partial \beta} \log L(\beta) \\ &= \sum_{t=1}^T \sum_{i=1}^n \left[ z_{t(i)} - \frac{\sum_{j=i}^{I_t} z_{t(j)} \exp(\beta' z_{t(j)})}{\sum_{j=i}^{I_t} \exp(\beta' z_{t(j)})} \right] \\ &= \sum_{t=1}^T \sum_{i=1}^n [z_{t(i)} - E_i^{(1)}], \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} I(\beta) &= \frac{\partial^2}{\partial \beta^2} \log L(\beta) \\ &= \frac{\partial}{\partial \beta} S(\beta) \\ &= \sum_{t=1}^T \sum_{i=1}^n [E_i^{(2)} - (E_i^{(1)})^{\otimes 2}], \end{aligned} \tag{2.6}$$

where

$$E_i^{(1)} = \frac{\sum_{j=i}^{I_t} z_{t(j)} \exp(\beta' z_{t(j)})}{\sum_{j=i}^{I_t} \exp(\beta' z_{t(j)})}, \quad E_i^{(2)} = \frac{\sum_{j=i}^{I_t} z_{t(j)}^{\otimes 2} \exp(\beta' z_{t(j)})}{\sum_{j=i}^{I_t} \exp(\beta' z_{t(j)})},$$

and  $A(p \times q) = (a_{ij})$ , will result in  $A^{\otimes 2}(p \times q) = (a_{ij}^2)$ .

The  $S(\beta)$  here is used as the function for solving  $\beta$ . Plugging in the Newton-Raphson method, we establish the following procedure:

For the  $k^{th}$  iteration, the solution of  $\beta$  is given by

$$\beta^{(k)} = \beta^{(k-1)} - [I(\beta^{(k-1)})]^{-1} S(\beta^{(k-1)}), \quad (2.7)$$

the iteration terminates at the  $k^{th}$  iteration if  $S(\beta^{(k-1)})$  is close enough to zero or the difference between  $\beta^{(k)}$  and  $\beta^{(k-1)}$  is negligible.



# Chapter 3

## Random Effects

From the previous chapter, we have developed a semi-parametric regression model to solve for the problem on modeling ranked data. However, the multinomial logit model alone is unable to explain the unobserved correlated characteristics of clusters in the ranked data.

To model the correlation within a cluster in the field of life-time data, frailty models are developed (cf. Gu, Sun and Huang, 2002). This is a common approach in the modeling of cluster correlations such that frailties are introduced as an unobservable latent variable for relative clusters. However, frailty models cannot be applied directly to our rank regression model. It is because we are considering multistage ranked data, which is slightly different from the life-time data. As a modification, we employ random effects, which is another latent variable approach, similar to the frailty models.

### 3.1 Definition

Laird and Ware introduced the random effects in 1982, for the analysis of longitudinal data. This has been used to store the effects of repeated measurements

on the same individual, and had a wide range of applications in medical studies, in which the measurements might be blood pressure, cholesterol level, and others. These measurements are taken for different person, where each person might obtain one or more of the same kind of measurement over time. The random effects represent the correlations of the same person's measurements. Therefore, if we view each person as a cluster, the random effects can also be used in the model of rank regression proposed.

Random effects models have several desirable features. They allow modeling and analysis of between and within clusters variations at the same model. Furthermore, the random effects parameters usually have a natural interpretation relevant to the goals of the study. In our data concerned, the random effects accounts for the quality of each cluster.

To be more specific, let  $\mu_h$  denote the unobserved latent variable or the random effects for cluster  $h$ , in which object  $i$  belongs to. The utility of object  $i$  in chapter 2 (2.1) will then be modified as:

$$U_i = \beta' z_i + \mu_h + \epsilon_i \quad (3.1)$$

the random effects  $\mu_h (h = 1, \dots, H)$ , for the total number of  $H$  clusters, are assumed to be independent and identically distributed random variables with a normal distribution of  $N(0, \sigma^2)$ . We assume  $\sigma^2$  is known in this thesis.

This utility is combined with the multinomial logit model, and the resulting probability expression for win will be:

$$P_i = \frac{\exp(\beta' z_i + \mu_{h(t,i)})}{\sum_{j=1}^I \exp(\beta' z_j + \mu_{h(t,j)})} \quad (3.2)$$

where  $h(t, i)$  indicates the cluster, in which the object having rank  $i$  in the  $t^{th}$  comparison belongs to.



### 3.1.1 A Simple Estimation Algorithm

In order to demonstrate the idea of random effects, we construct a simple estimation algorithm by using a simple linear regression model on a transformed version of the ranked data. The model is considered to be:

$$Y_i = \beta' z_i + \mu_h + \epsilon_i$$

instead of the utility  $U_i$  in the semi-parametric model, it is replaced by a rough response variable  $Y_i$ , which is given by:

$$Y_i = 0.5 - \frac{rank_i}{I+1}$$

it can be seen that  $Y_i$  is a scaled and centered variable which serve as a substitute to the performance of object  $i$  by its rank in a comparison.

By using statistical computing packages such as S-plus, we can easily obtain estimates of the regression parameters  $\hat{\beta}$  and residuals  $R_i$ , given by:

$$R_i = Y_i - \hat{\beta}' z_i$$

Then, we can obtain a rough estimate of the random effects by the residuals specific to each cluster:

$$\mu_h = \frac{1}{I_h} \sum_{i=1}^{I_h} R_{h(i)}$$

where  $h(i)$  indicates the  $i^{th}$  observation that belongs to cluster  $h$  and  $I_h$  is the total number of observations for cluster  $h$ .

We apply this simple estimation algorithm to the horse racing data used in chapter 4. Details of the data set will not be discussed here. A histogram of the random effects found is plotted in Figure 3.1 for a rough concept of random

effects. From the figure, we can see that the random effects has a rather normal distribution, which satisfies our distribution assumption above.

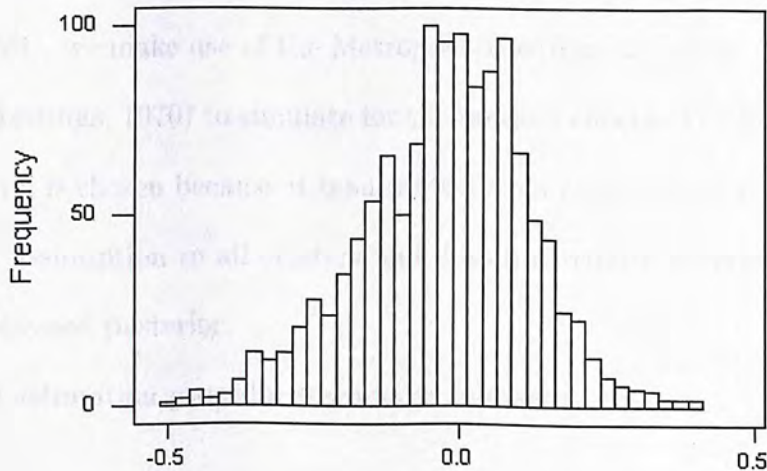


Figure 3.1: Plot of the random effects by simple estimation algorithm

### 3.2 Metropolis-Hastings Algorithm for Simulating Random Effects

In order to obtain estimates for the random effects, the Newton-Raphson method is not applicable as the number of clusters is very large. There is a computational difficulty and it is almost impossible to construct an explicit form of the information matrix from the likelihood concerned. Therefore, we try to employ simulation algorithm to obtain the random effects from its posterior distribution:

$$f(\mu) \propto \prod_{t=1}^T \prod_{i=1}^n \left\{ \frac{\exp(\beta' z_{t(i)} + \mu_{h(t,i)})}{\sum_{j=i}^{I_t} \exp(\beta' z_{t(j)} + \mu_{h(t,j)})} \right\} \left\{ \prod_{h=1}^H \frac{1}{\sigma} \phi\left(\frac{\mu_h}{\sigma}\right) \right\}, \quad (3.3)$$

where  $\phi(\cdot)$  stands for the density function of  $\mu_h$ .



Moreover, direct generation of random effects from this posterior (3.3) remains a very difficult task. Therefore, Markov Chain Monte Carlo (MCMC) was chosen as a powerful means for generating random samples to be used in computing further statistical estimates. Among the two most popular specifications in MCMC, we make use of the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) to simulate for the random effects. The Metropolis-Hastings algorithm is chosen because it is suitable for our condition of a common posterior density assumption to all clusters and does not require generating samples from a complicated posterior.

The estimation procedures works as follows:

For the  $h^{th}$  cluster in the  $k^{th}$  iteration,

(a) We take  $\mu_h = \sigma\nu_h$ , generate  $\nu_h^*$  from  $N(0, 1)$ .

(b) Calculate  $\alpha_h = \min\{1, \alpha_h^*\}$ , where

$$\alpha_h^* = \frac{\prod_{t=1}^T \left[ \prod_{i=1}^n \left\{ \frac{\exp(\beta' z_{t(i)} + \sigma\nu_{h(t,i)}^*)}{\sum_{j=i}^{I_t} \exp(\beta' z_{t(j)} + \sigma\nu_{h(t,j)}^*)} \right\} 1_{[h \in \Theta_t]} \right]}{\prod_{t=1}^T \left[ \prod_{i=1}^n \left\{ \frac{\exp(\beta' z_{t(i)} + \sigma\nu_{h(t,i)})}{\sum_{j=i}^{I_t} \exp(\beta' z_{t(j)} + \sigma\nu_{h(t,j)})} \right\} 1_{[h \in \Theta_t]} \right]}$$

(c) Generate  $u$  from uniform(0, 1),

if  $u \leq \alpha_h$ , set  $\nu_h^{(k)} = \nu_h^*$ ; otherwise  $\nu_h^{(k)} = \nu_h^{(k-1)}$ .

However, the random effects generated by these procedures will not converge to a single value due to the random number generation. Therefore, there is a need for a further development of estimation algorithm to facilitates the parameter estimation on the complete model concerned.

### 3.3 EM Algorithms for Maximizing the Likelihood

The problem left is to obtain estimates for the random effects  $\mu$  in alignment with the rank regression parameters  $\beta$  by maximizing the likelihood:

$$L(\beta) = \int \dots \int \prod_{t=1}^T \prod_{i=1}^n \left\{ \frac{\exp(\beta' z_{t(i)} + \mu_{h(t,i)})}{\sum_{j=i}^{I_t} \exp(\beta' z_{t(j)} + \mu_{h(t,j)})} \right\} \left\{ \prod_{h=1}^H \frac{1}{\sigma} \phi\left(\frac{\mu_h}{\sigma}\right) d\mu_h \right\} \quad (3.4)$$

However, direct maximization for the above likelihood function is still impossible even with the aid of fast computers. Therefore, the EM algorithm (Dempster et. al., 1977) is chosen for the general approach to compute maximum likelihood estimates iteratively for incomplete data. It involves an Estimation (E-) step and a Maximization (M-) step. This process is preferable of its simplicity as it allows simulation of parameters in its E-step and does not require the maximization of complicated likelihood in its M-step.

In our model concerned, we simulate sets of random effects  $\mu$  by the Metropolis-Hastings algorithm in the E-step. With these sets of  $\mu$ , instead of maximizing the likelihood (3.4), we simply need to compute estimates for the parameters  $\beta$  in the M-step by maximizing the partial likelihood  $L(\beta|\mu)$ , given by:

$$L(\beta|\mu) = \prod_{t=1}^T \prod_{i=1}^n \left\{ \frac{\exp(\beta' z_{t(i)} + \mu_{h(t,i)})}{\sum_{j=i}^{I_t} \exp(\beta' z_{t(j)} + \mu_{h(t,j)})} \right\}$$

to maximize partial likelihood of this form, the Newton-Raphson method defined in chapter 2 (2.7) can be applied directly.

Two versions of the EM algorithms, namely Stochastic EM (SEM) and Monte Carlo EM (MCEM) algorithms are introduced for alternative comparisons.



### 3.3.1 Stochastic EM Algorithm

The Stochastic EM (SEM) algorithm (Celeux and Diebolt, 1985) was the first stochastic version of EM algorithm developed. Note that SEM algorithm is different from the Stochastic Approximation EM (SAEM) algorithm (cf. Delyon et. al., 1999), which is commonly used recently. The SEM algorithm is chosen for its easy computational steps. It involves the generation of a sequence of maximum likelihood estimates. An estimate of the sequence is defined as the SEM estimator.

The SEM algorithm works for our model as follows:

In the  $k^{th}$  iteration, given the current  $\beta^{(k-1)}$ ,

(a) generate a set of  $\mu^{(k)}$  by the Metropolis-Hastings algorithm developed in section 3.2.

(b) estimates  $\beta^{(k)}$  by the Newton-Raphson method on  $L(\beta|\mu)$ .

The iteration will be repeated for  $m$  times.

Although the sets of  $\beta$  and  $\mu$  does not converge, we obtain the SEM estimators by:

$$\hat{\beta} = \frac{1}{m - m_0} \sum_{k=m_0}^m \beta^{(k)}, \quad \hat{\mu} = \frac{1}{m - m_0} \sum_{k=m_0}^m \mu^{(k)}$$

where  $m_0$  is the length of the 'burn-in' period in order to reduce the influence of the initial conditions and unstable fluctuations. In our procedures,  $m_0$  is taken to be 80% of  $m$ .

We point out here that this algorithm does not converge to a maximum of likelihood (3.4). But in practice, the solution is close to what has been obtained by the MCEM algorithm.



### 3.3.2 MCEM Algorithm

In this section, we are going to perform the EM algorithm through the method of Monte Carlo (Wei and Tanner, 1990).

The algorithm acts as follows. In the  $k^{th}$  iteration, given the current  $\beta$ ,

(a) generate a sample of random effects sets  $\mu_{(1)}, \dots, \mu_{(m_k)}$  by the Metropolis-Hastings algorithm.

(b) calculate  $I_{(i)}(\beta)$  and  $S_{(i)}(\beta)$  for each set of  $\mu_{(i)}$  on  $L(\beta|\mu)$ . Instead of using each  $I_{(i)}(\beta)$  and  $S_{(i)}(\beta)$  to solve for  $\beta_{(i)}^{(k)}$ , we estimate  $I(\beta)$  and  $S(\beta)$  by:

$$I(\beta) = \frac{1}{m_k} \sum_{i=1}^{m_k} I_{(i)}(\beta), \quad S(\beta) = \frac{1}{m_k} \sum_{i=1}^{m_k} S_{(i)}(\beta)$$

(c) we then estimates  $\beta^{(k)}$  with  $I(\beta)$  and  $S(\beta)$ . That is,

$$\beta^{(k)} = \beta^{(k-1)} - \left[ \frac{1}{m_k} \sum_{i=1}^{m_k} I_{(i)}(\beta^{(k-1)}) \right]^{-1} \left[ \frac{1}{m_k} \sum_{i=1}^{m_k} S_{(i)}(\beta^{(k-1)}) \right],$$

and in the last iteration considered, we estimates the random effects by the  $m_k$  results of  $\mu_{(i)}$ 's.

$$\hat{\mu} = \frac{1}{m_k} \sum_{i=1}^{m_k} \mu_{(i)}$$

Two important considerations in regarding the implementation of MCEM algorithm are monitoring the convergence of the algorithm and the specification of  $m_k$ . In specifying  $m_k$ , it is unnecessary to start with a large value of  $m_k$  as the current approximation to the maximizer may be far from the true value. It is useful to increase  $m_k$  as the current approximation moves closer to the true maximizer. After a certain number of iterations, the process will be stabilized, and the desired results can be found. Therefore, we take several iterations and with the value of  $m_k$  increases progressively. In our procedures, we consider 10 iterations with  $m_k$  increases from 10 ( $k = 1$ ) to 1200 ( $k = 10$ ).

By the simulation and calculation algorithms introduced above, we will be able to obtain parallel sets of regression parameters  $\beta$  and the random effects  $\mu$ . These techniques will be illustrated in the next chapter for an application to a real data set.

## Chapter 4

### Application

In this chapter, we apply the semi-parametric model with random effects to a horse racing data set collected from the winter 2001 to February 2002 comprised of races held in the Shatin Racecourse organized by the racing horse Jockey Club. The data set contains 721 races, in which all the international and special races are excluded. We use the data relevant to the first 500 races to compute the estimates of the parameter  $\beta$  and random effects  $\mu$  for 1212 horses. Then, the estimates are used to calculate the winning probabilities of the remaining 221 races, and the results are used to construct betting strategies to justify the structure of the model.

In horse racing, only the horses finishing in the first five positions are awarded prizes. The jockey of a horse with no prize is called a loser, and the horse is called a loser as hard as for the jockey. The horses with no prize are called the "losers" and the jockey is called the "loser". The horses with no prize are called the "losers" and the jockey is called the "loser". The horses with no prize are called the "losers" and the jockey is called the "loser".



## Chapter 4

# Application

In this chapter, we apply the semi-parametric model with random effects to a horse racing data set collected from December 2000 to February 2003 comprised of races held in the Shatin Racecourse conducted by the Hong Kong Jockey Club. The data set contains 721 races, in which all the international and special races are excluded. We use the data relevant to the first 500 races to compute the estimates of the parameter  $\beta$  and random effects  $\mu$  for 1263 horses. Then, the estimates are used to calculate the winning probabilities of the remaining 221 races, and the results are used to construct betting strategies to justify the accuracy of the model.

In horse racing, only the horses finishing in the first few positions are awarded prizes. The jockey of a horse with no hope of getting a prize may not drive the horse as hard as he/she can. That is, the order of finish beyond the first few places may not reflect the true strength of those horses. Therefore, we choose  $n$  equals to 4 in the likelihood function (3.4) for simplicity. That is, the likelihood function becomes:

$$L(\beta) = \int \dots \int \prod_{t=1}^T \prod_{i=1}^4 \left\{ \frac{\exp(\beta' z_{t(i)} + \mu_{h(t,i)})}{\sum_{j=i}^{I_t} \exp(\beta' z_{t(j)} + \mu_{h(t,j)})} \right\} \left\{ \prod_{h=1}^H \frac{1}{\sigma} \phi\left(\frac{\mu_h}{\sigma}\right) d\mu_h \right\}$$



## 4.1 Fundamental Variables and Variable Selection

For each horse in a race, we start with considering 46 predictor variables. The inclusion of these variables does not mean that these are the only variables that are important for the prediction. Among these 46 variables, we try to select significant variables in the semi-parametric regression model before fitting in the random effects.

For each of the 46 variables,

(a) find estimates of the parameters  $\hat{\beta}$  and standard errors  $\widehat{SE}(\hat{\beta})$  from the multinomial logit model given in chapter 2;

(b) construct a t-value for each variable, in which

$$t_i = \frac{\hat{\beta}_i}{\widehat{SE}(\hat{\beta}_i)}$$

(c) by a simple hypothesis testing, we reject the variables with  $|t_i| < T_0$ .

In order to avoid eliminating factors that should be significant but affected by insignificant factors, we repeat the selection procedures for a few times. A loose  $T_0$  is imposed in the beginning, such that  $T_0^{(1)} = 1.00$  and is tightened until  $T_0 = 1.96$ . Finally, we select 14 variables with t-values larger than 1.96.

The maximum likelihood estimates  $\hat{\beta}$  of the regression parameter for these 14 variables, the corresponding standard errors and their t-values are shown in Table 4.1. Specification of these predictor variables can be found in the appendix.

**LogOdds** is included in the 14 fundamental variables of our model and from Table 4.1, it is the most important variable in our model. It has been shown by Gu, Huang and Benter (2002) that this factor is a rather accurate estimate of

the winning probabilities, but due to the pari-mutuel system, it cannot provide profitable strategies as there is a 17.5% track take by the Jockey Club. Therefore, we include this factor and try to obtain a better estimate of the winning probabilities with the other factors.

It should also be noted that the the factor **DaySince** and the variables of its transformation are chosen as our fundamental variables. The motivation of introducing **DaySince** is that, when we are in a trainer's point of view, we will only allow the horse to participate frequently if it is in a very good condition. Therefore, it is important to take this factor into account.

Table 4.1: The MLE and SE for selected variables under the multinomial logit model

No.	Variable	$\hat{\beta}$	$\overline{SE}(\hat{\beta})$	$t_i$
1	LogOdds	-0.7578	0.0508	-14.9173
2	WtCarried	-8.5555	4.0427	-2.1163
3	WtCarriedByLogDist	0.0113	0.0056	2.0179
4	LogHWgt	6.0313	2.1150	2.8517
5	LogWeightByDist	-0.0042	0.0014	-2.9866
6	AgeT	-1.5765	0.5596	-2.8172
7	DaySince	0.0287	0.0039	7.4082
8	SqrtDaySince	-1.8320	0.7902	-2.3184
9	LogDaySince	2.1263	0.2891	7.3549
10	LogDaySinceT	0.8671	0.1862	4.6568
11	LogHWgtChg	5.0329	2.3655	2.1276
12	SqrtLHNR	-0.2965	0.0604	-4.9089
13	LogLHNR	0.5483	0.1081	5.0722
14	AveStdRank	-0.3333	0.1317	-2.5308



## 4.2 Simulation Results

By applying the parameter estimation developed in the previous chapters, we obtain two sets of parameters  $\beta$  and random effects  $\mu$  relative to the SEM and MCEM algorithm. The estimates of the regression parameters  $\beta$  for the selected variables by the two algorithms are shown in Table 4.2. The histogram of the two sets of random effects are plotted in Figures 4.1 and 4.2.

Table 4.2: The MLE and SE for selected variables by SEM and MCEM algorithms

No.	Variable	SEM algorithm		MCEM algorithm	
		$\hat{\beta}$	$SE(\hat{\beta})$	$\hat{\beta}$	$SE(\hat{\beta})$
1	LogOdds	-0.7921	0.0437	-0.7864	0.0498
2	WtCarried	-9.0217	4.0032	-9.0308	3.9923
3	WtCarriedByLogDist	0.0138	0.0057	0.0141	0.0053
4	LogHWgt	8.2926	2.8885	8.1793	2.6545
5	LogWeightByDist	-0.0053	0.0014	-0.0051	0.0013
6	AgeT	-1.7815	0.5487	-1.7729	0.5168
7	DaySince	0.0349	0.0056	0.0357	0.0048
8	SqrtDaySince	-2.0015	0.7653	-0.1982	0.7821
9	LogDaySince	2.3148	0.3516	2.2871	0.3408
10	LogDaySinceT	0.9342	0.2054	0.9648	0.1996
11	LogHWgtChg	6.3826	2.6851	6.3729	2.4173
12	SqrtLHNR	-0.3129	0.0784	-0.3038	0.0687
13	LogLHNR	0.6610	0.1574	0.6271	0.1267
14	AveStdRank	-0.3563	0.1309	-0.3683	0.1382

From the two graphs, we can see that similar pattern appeared. The two sets of random effects seemed to follow a normal distribution. As the number of races



participated for each horse is different, the assumption of normal distribution in the generation of random effects is reasonable.

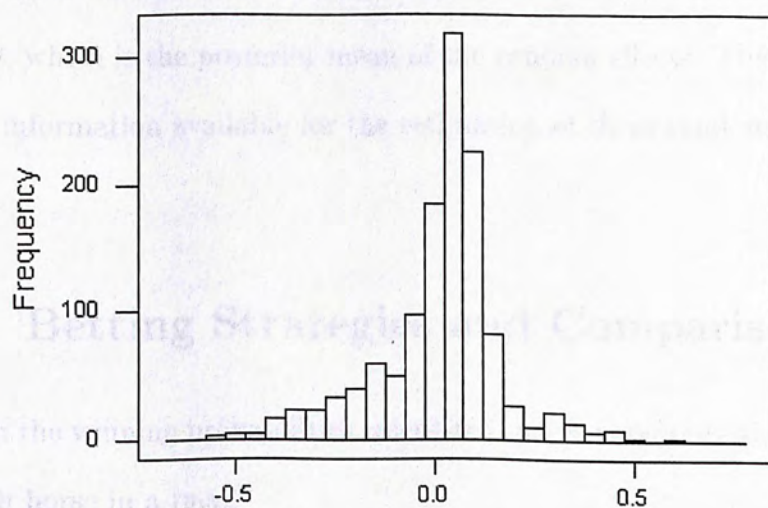


Figure 4.1: Plot of the random effects by SEM algorithm

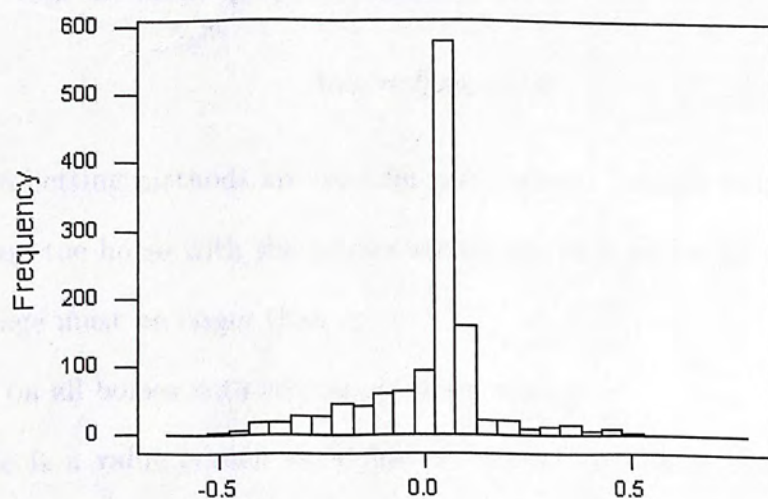


Figure 4.2: Plot of the random effects by MCEM algorithm

Using the simulated results, we can construct the winning probabilities  $P_i$  for each horse. Recall equation (3.3):

$$P_i = \frac{\exp(\beta' z_i + \mu_{h(t,i)})}{\sum_{j=1}^I \exp(\beta' z_j + \mu_{h(t,j)})}.$$

For those horses which has not been raced before, the random effects are taken to be 0, which is the posterior mean of the random effects. This is because there are no information available for the estimation of these random effects.

### 4.3 Betting Strategies and Comparisons

With the winning probabilities calculated, we can compute the expected return for each horse in a race:

$$Exp_i = Odds_i \times P_i$$

After computing this expectation, we can find the advantage that can be gained from the model proposed for each horse:

$$Adv_i = Exp_i - 1.0$$

Two betting methods are used for comparison. 1 dollar is bet:

(i) on the horse with the largest advantage in a particular race, in which its advantage must be larger than  $c$ .

(ii) on all horses with advantage larger than  $c$ .

where  $c$  is a value chosen such that we expect to obtain better results in the profit. In this example,  $c$  is chosen to be 0.15.

However, we should also be aware of some horses that has very small values in its estimated winning probability, but obtain a large advantage simply due to



the large value of its payoff odds (That is, the public do not expect this horse to win). Also, we should note that for these large odds horses, we do not have enough data to accurately estimate their win probability. It is because for all horses that has an odds larger than 100, the Jockey Club would announce their odds as 99. In order to avoid betting on these horses, we impose a restriction on our betting such that horses with odds larger than a certain amount will not be betted, and is taken to be 40 here.

For a comparison between models with and without random effects, we construct betting strategies by using estimates of the parameter  $\beta$  from Table 4.1 (without random effects), and the estimates of  $\beta$  with random effects  $\mu$  by the two EM algorithms. The betting results are shown in Table 4.3.

Table 4.3: Results for simple betting strategy in 221 races

	Betting method(i)			Betting method(ii)		
	No ran- dom effects	SEM algorithm	MCEM algorithm	No ran- dom effects	SEM algorithm	MCEM algorithm
No. of bets made	178	189	190	347	381	395
No. of races won	12	21	20	21	33	35
Amounts bet (\$)	178	189	190	347	381	395
Profit (\$)	-12.30	19.30	15.50	5.30	34.10	37.70
Returns rate	-0.0691	0.1021	0.0816	0.0153	0.0895	0.0954

From the results, we can observe that the semi-parametric models with random effects perform better than the model without random effects overall. This



can be seen from both the improved returns rate, and also the larger number of races won to races bet ratio in the latter models. Therefore, we can conclude that improvements are made by the random effects.

In addition to the simple wagering strategy employed above, we also adopt the Kelly strategy (Kelly, 1956; Rosner, 1975; MacLean, Ziemba and Blazenko, 1992; Benter, 1994; Gu, Huang and Benter, 2002). In this strategy, the amount of bet each time is not simply 1 dollar as in the former betting strategy. Instead, it is taken to be:

$$A = K \times \frac{\text{Advantage}}{\text{Odds} - 1.0}$$

where  $K$  is the total capital of an investor, which would vary from one investment to the other. We applied a modified (simpler) Kelly strategy as in Gu, Huang and Benter (2002) to the models with random effects, in which  $K$  is fixed at a constant 1000. Keeping all other conditions the same as the simple wagering strategy, the new results are shown in the Table 4.4.

It may be argued that the return rate in our betting strategies are not very large, but one should be reminded that there is a track take of expected rate -0.175 imposed before our modeling. Therefore, if the resulting returns rate is larger than -0.175, the results can be considered as successful.

## Chapter 5

## Conclusions and Further Studies

Table 4.4: Results for Kelly strategy in 221 races

	Betting method(i)			Betting method(ii)		
	No ran- dom effects	SEM algorithm	MCEM algorithm	No ran- dom effects	SEM algorithm	MCEM algorithm
No. of bets made	178	189	190	347	381	395
No. of races won	12	21	20	21	33	35
Amounts bet (\$)	4537.29	4876.39	4982.67	6058.14	6734.28	6877.45
Profit (\$)	40.71	273.02	53.91	153.75	206.14	181.32
Returns rate	0.0090	0.0560	0.0108	0.0254	0.0414	0.0264

## Chapter 5

# Conclusions and Further Studies

The main aim in this thesis is to introduce a new factor called the random effects into the rank regression model and establish MCMC methods to get estimates of regression parameters. From the application example, it has been shown that the models with random effects perform better than the model without random effects for clustered data as suggested by Longford (1993). Therefore, we can conclude that with random effects, rank regression model is improved. This implies that when we are considering rank regression model, the cluster effect is also an important component of the model and should not be ignored.

It should be noted that in our analysis, we have used the payoff odds as one of the fundamental variables in our estimation. This variable cannot be used to make bets in real-life since it can only be obtained when all the bets have been made. However, the Jockey Club always provides pre-races odds information continuously. Therefore, we may use a variable that is the odds announced right before the races as a close substitute for the final odds. Replacing the odds by a substitute may change the performance of the model, but the size of the change should not affect our conclusions.



The results found in the application example are obtained from the prediction on 221 races. Although this number of races may not be large enough to provide a very detailed outcome, and to construct reliable betting strategies. It is sufficient to support our main idea of improvements on models by the random effects. A more conclusive result can be obtained with a larger data set with more races informations.

In recent studies, it is found that the probit model performs better in the analysis of rank regression, especially for the horse-racing data. However, it is a more complex model and requires more time in the computation of the output. Therefore, it was not chosen for the study of random effects on rank regression models. As it has been proved in this thesis that the random effects makes improvements on rank regression models, it is worthy to progress the study for rank regression model with random effects in other models.

Besides horse racing data, the model found and the estimation algorithm developed in this thesis have many other possible applications. We may try to apply this model to clinical data, academic data, economic data, and other ranked data to estimate their rank probabilities for further analysis.

# Appendix

Specification of the 46 predictor variables considered in the horse racing data set in chapter 4:

1. **LogOdds**: log of the payoff odds in the race;
2. **WtCarried**: weight carried by the horse in the race;
3. **WtCarriedByDist**: the **WtCarried** times the distance of the race;
4. **WtCarriedByLogDist**: the **WtCarried** times the log of the distance of the race;
5. **LogHWgt**: log of the horse weight;
6. **LogWeightByDist**: the **LogHWgt** times distance of the race;
7. **AgeT**: a nonlinear transformation of horse's age;
8. **AgeTByDist**: the **AgeT** times the distance of the race;
9. **DaySince**: number of days since last race;
10. **SqrtDaySince**: square root of **DaySince**;
11. **LogDaySince**: log of **DaySince**;
12. **LogDaySinceT**: transformation of **LogDaySince**;



13. **RALogDaySByTPlacePer**: race average of **LogDaySince** times trainer's place percentage;
14. **HWgtChg**: horse weight change;
15. **LogHWgtChg**: log of **HWgtChg**;
16. **RatingT**: transformation of the rating;
17. **HPlacePer**: horse place percentage;
18. **LHNR**: the number of races participated by the horse + 1;
19. **SqrtLHNR**: square root of **LHNR**;
20. **LogLHNR**: log of **LHNR**;
21. **AlmostNewH**: indicator for horse race less than 4 times;
22. **LastLogOdds**: the log of final odds for the horse's last race;
23. **AveStdNLBHChg**: average standard no. of lack behind change;
24. **AveStdNLBHChgByLastLogOdds**: the **AveStdNLBHChg** times **LastLogOdds**;
25. **WgtAveStdNLBH**: weighted average of the standard no. of lack behind;
26. **NewDist**: indicator variable for difference more than 200m in distance of the race and the average of the past races;
27. **NewDist2**: indicator variable for difference more than 400m in distance of the race and the average of the past races;
28. **NewDist2ByLogAge**: the **NewDist2** times log of the horse's age;

29. **NewST**: indicator variable for first time running in Shatin Racecourse;
30. **AveStdRank**: average standard rank of the horse in past 20 races;
31. **AveStdRankChg**: average standard rank change;
32. **JPlacePer**: jockey's place percentage;
33. **JWinPerChg**: jockey's quality change;
34. **StdDraw**: standard initial post position;
35. **StdDrawByRaceNoT**: the **StdDraw** times the **RNT** (race no. on the turf);
36. **StdDrawByRNTByGoing**: the **StdDraw** times the **RNT** times the **Going** (an indicator variable for the quality of the turf: from 1 - dry and fast, to 8 - wet and muddy);
37. **StdDrawByGoing**: the **StdDraw** times the **Going**;
38. **StdDrawByDistTByCourse**: the **StdDraw** times the **Course** (an indicator variable, in which 0 for "A" course, 1 for "B" course, 2 for "C" course and 2.75 for "C+3" course);
39. **StdDrawByAveRatingByGoing**: the **StdDraw** times the average rating times the **Going**;
40. **StdDrawByAveRatingByCourse**: the **StdDraw** times the average rating times the **Course**;
41. **AveRatByAveStdNLBHChg**: the average rating times the **AveStdNLBHChg**;



42. **AveRatByLogAgeByWtCarried**: the average rating times the log of the horse's age times the **WtCarried**;
43. **SqrtLHNRByRAHWinPer**: square root of the **LHNR** times the race average of the horse's win percentage;
44. **SqrtLHNRByAveStdRank**: square root of the **LHNR** times the **AveStdRank**;
45. **SqrtDaySinceBySeason**: the **SqrtDaySince** times the **Season** (indicator variable, in which 0 for beginning of racing season, and 1 for end of racing season);
46. **SqrtLHNRBySeason**: square root of the **LHNR** times the **Season**.

# Bibliography

1. Aitkin, M., Anderson, D. and Hinde, J. (1981). Statistical modelling of data on teaching styles. *Journal of the Royal Statistical Society, Series A*, **144**, 419-448.
2. Ali, M.M. (1998). Probability Models on Horse-race Outcomes. *Journal of Applied Statistics*, **25**, 221-229.
3. Bennett, S. (1983). Analysis of survival data by the proportional odds model. *Statistics in Medicine*, **2**, 273-277. Wiley, New York.
4. Benter, W. (1994). Computer Based Horse Race Handicapping and Wagering Systems: A Report. in *Efficiency of racetrack betting Markets*, D.B. Hausch, V.S.Y. Lo, and W.T. Ziemba eds., San Diego: Academic Press, 183-198.
5. Bolton, R.N. and Chapman R.G. (1986). Searching for Positive Retruns at the Track: A Multinomial Logit Model for Handicapping Horse Races. *Management Science*, **32**, 1040-1060.
6. Boskin, M.J. (1974). A Conditional Logit Model of Occupational Choice. *Journal of Political Economy*, **82**, 389-398.



7. Carnahan, B., Luther, H.A. and Wilkes, J.O. (1969). *Applied Numerical Methods*. John Wiley, New York.
8. Celeux, G. and Diebolt, J. (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistical Quarterly*, **2**, 73-82.
9. Chapman, R.G. (1979). Pricing Policy and the College Choice Process. *Research in Higher Education*, **10**, 37-57.
10. Chapman, R.G. (1980). Retail Trade Area Analysis: Analytics and Statistics. in *Proceedings: Market Measurement and Analysis*, Robert A. Leone, ed. Providence, RI: TIMS College on Marketing and The Institute of Management Sciences, 40-49.
11. Chapman, R.G. and Staelin, R. (1982). Exploiting Rank Ordered Choice Set Data Within the Stochastic Utility Model. *Journal of Marketing Research*, **19**, 288-301.
12. Cox, D.R. (1972). Regression Models and Life-Tables. *Journal of the Royal Statistical Society, Series B*, **34**, 187-220.
13. Cuzick, J. (1988). Rank Regression. *The Annals of Statistics*, **16**, 1369-1389.
14. Delyon, B., Lavielle, M. and Moulines, E. (1999). Convergence of a Stochastic Approximation Version of the EM Algorithm. *The Annals of Statistics*, **27**, 94-128.
15. Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal*

16. Domencich, T. and McFadden, D. (1975). *Urban Travel Demand: A Behavioral Analysis*. Amsterdam: North-Holland Publishing Company.
17. Everitt, B.S. (1993). *Cluster Analysis* (3rd ed.). John Wiley & Sons, New York.
18. Gensch, D.H. and Recker, W.W. (1979). The Multinomial, Multiattribute Logit Choice Model. *Journal of Marketing Research*, **16**, 124-132.
19. Green, P.E., Frank, R. E. and Robinson, P.J. (1967). Cluster analysis in test market selection. *Management Science*, **13**, 387-400.
20. Gu, M.G., Huang, C. and Benter W. (2002). Probit Models for Handicapped Horse Racing. Manuscript.
21. Gu, M.G., Sun, L. and Huang, C. (2002). A Universal Procedure for Parametric Frailty Models. Manuscript.
22. Hastings, W.K. (1970). Monte Carlo Sampling Methods Using Markov Chains and their Applications. *Biometrika*, **57**, 97-109.
23. Heckman, J. and Singer, B. (1984). A method for minimizing the impact of distributional assumptions in econometric models for duration data. *Econometrica*, **52**, 271-319.
24. Hodson, F.R. (1971). Numerical typology and prehistoric archaeology. in *Mathematics in the Archaeological and Historical Sciences*. Hodson, F.R., Kendall, D.G. and Tautu, P.A., eds., Edinburgh: University Press.

25. Jolliffe, I.T., Jones, B. and Morgan, B.J.T. (1982). Utilising clusters: a case study involving the elderly. *Journal of the Royal Statistical Society, Series A*, **145**, 224-236.
26. Joskow, P.L. and Mishkin, F.S. (1977). Electric Utility Fuel Choice Behavior in the United States. *International Economic Review*, **18**, 719-736.
27. Kelly, J. (1956). A New Interpretation of Information Rate. *Bell System Technical Journal*, **35**, 917-926.
28. Kohn, M.G., Manski, C.F. and Mundel, D.S. (1976). An Empirical Investigation of Factors Which Influence College-Going Behavior. *Annals of Economic and Social Measurement*, **5**, 391-419.
29. Laird, N.M. and Ware, J.H. (1982). Random-Effects Models for Longitudinal Data. *Biometrics*, **38**, 963-974.
30. Lancaster, T. and Nickell, S. (1980). The analysis of re-employment probabilities for the unemployed. *Journal of the Royal Statistical Society, Series A*, **143**, 141-165.
31. Li, M.M. (1977). A Logit Model of Homeownership. *Econometrica*, **45**, 1081-1097.
32. Longford, N.T. (1993). *Random Coefficient Models*. Oxford University Press, New York.
33. Luce, R.D. (1959). *Individual Choice Behavior*. Wiley: New York.
34. Luce, R. and Suppes, P. (1965). Preference, Utility, and Subjective Probability. in *Handbook of Mathematical Psychology*, R. Luce, R. Bush, and E.



- Galanter eds., New York: Wiley & Sons, Vol. 3, 249-410.
35. MacLean, L.C., Ziemba, W.T. and Blazenko, G. (1992). Growth Versus Security in Dynamic Investment Analysis. *Management Science*, **38**, 1562-1585.
  36. Marden, J.I. (1995). *Analyzing and modeling rank data*. Chapman & Hall, London.
  37. McFadden, D. (1974). Conditional Logit Analysis of Qualitative Choice Behavior. in *Frontiers in Econometrics*, P. Zarembka ed., New York: Academic Press, 303-328.
  38. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, **21**, 1087-1092.
  39. Paykel, E.S. and Rassaby, E. (1978). Classification of suicide attempters by cluster analysis. *The British Journal of Psychiatry*, **133**, 42-52.
  40. Pilowsky, I., Levine, S. and Boulton, D.M. (1969). The classification of depression by numerical taxonomy. *The British Journal of Psychiatry*, **115**, 937-945.
  41. Plackett, R.L. (1975). The analysis of Permutations. *Applied Statistics*, **24**, 193-202.
  42. Punj, G.N. and Staelin, R. (1978). The Choice Process for Graduate Business Schools. *Journal of Marketing Research*, **15**, 588-598.

43. Ralston, A. and Wilf, H.S. (1966). *Mathematical Methods for Digital Computers*, Vol. 2. Wiley, New York.
44. Rosner, B. (1975). Optimal Allocation of Resources in a Pari-Mutuel Setting. *Management Science*, **21**, 997-1006.
45. Vergin, R.C. (1977). An Investigation of Decision Rules for Thoroughbred Race Horse Wagering. *Interfaces*, **8**, 34-45.
46. Wastell, D. G. and Gray, R. (1987). The numerical approach to classification: a medical application to develop a typology for facial pain. *Statistics in Medicine*, **6**, 137-164.
47. Wei, G.C.G. and Tanner, M.A. (1990). A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association*, **85**, 699-704.
48. Ziemba, W.T. and Hausch, D.B. (1984). *Beat the Racetrack*, Harcourt Brace Jovanovich Publishers, San Diego, Cal.







CUHK Libraries



004077288